## REMARKS

Attached hereto is an Excess Claims Fee letter for three excess total claims.

Claims 1-16 and 21-38 are all the claims presently pending in the application. New claims 36-38 are added.

It is noted that, notwithstanding any claim amendments made herein, Applicants' intent is to encompass equivalents of all claim elements, even if amended herein or later during prosecution.

Claims 27-35 stand rejected under 35 U.S.C. § 112, first paragraph, as failing to be enabled.

Claims 1-7 and 25-35 stand rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter.

Claims 1-16 and 21-35 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over IBM's article "LPEX User's Guide and Reference", further in view of IBM's article "Incrementally Imbedded Messages in an Edit View."

These rejections are respectfully traversed in the discussion below.

## I. THE CLAIMED INVENTION

Applicants' invention, as defined for example in the non-limiting embodiment of independent claim 1 (and substantially similarly in independent claims 8 and 13) is directed to a programmable text processing module which loads the document and a parsing editor for initially parsing the document and thereafter incrementally parsing changes committed in the document.

As providing an editing function that, in one exemplary embodiment does not modify the document in any manner (although this functionality is not precluded in variations of the present invention), the present invention provides "virtual" marks and associated action that typically exist only during the editing session (although, variations allow the metadata to be saved for subsequent/offline handling).

Therefore, in its role as an editor, the present invention includes also a mark control module that sets a plurality of marks in the document and provides a method for modifying the marks and clearing the marks. Each of the marks comprise selected information in the document and there is a means to link the selected information with a command, wherein the linking and setting is responsive to the operation of the parsing editor without user intervention.

New claims 36-38 focus on the data structure aspect of the present invention, as described beginning at line 3 on page 12 of the specification.

An exemplary configuration of an edit system incorporating the activemark structure of the present invention is shown in Fig. 1 of the application.

The conventional systems, such as those discussed below and in the Related Art section of the present application, do not have such a structure, and fail to provide for such an operation.

Indeed, such features are clearly not taught or suggested by the cited reference.

## II. THE REJECTION UNDER USC 35 §101

The Examiner alleges that claims 1-7 and 25-35 are directed to non-statutory subject matter because independent claim 1 "... refers to a software per se and are not tangibly embodied on a computer readable medium or hardware."

Applicants respectfully submit that the Examiner's position is faulty as a matter of law. As clearly stated in MPEP §2111: "*The broadest reasonable interpretation of the claims must also be consistent with the interpretation that those skilled in the art would reach.*"

In the present evaluation, the Examiner's interpretation of the claim language fails to recognize that the preamble clearly defines the scope of the claim as addressing "A <u>processing system</u> .... " Applicants submit that, to one of ordinary skill in the art, this term clearly conveys a concrete embodiment of the modules described in the claim limitations.

That is, although the Examiner may be entitled to consider that the exemplary embodiment described in the specification teaches the invention as embodied in software modules, the claim addresses the "<u>processing system" that incorporates these software modules</u>. In attempting to isolate these limitations outside the scope of the claim preamble, as including the transition word "comprising", the Examiner's interpretation fails to heed the plain meaning of the claim language, as that language would be interpreted by those skilled in the art.

Accordingly, Applicants respectfully request that the Examiner reconsider and withdraw the rejection alleging that these claims are addressed to non-statutory subject matter by reason of including limitations describing software modules.

## III. THE REJECTION UNDER 35 USC §112 , FIRST PARAGRAPH

The Examiner continues to allege that claims 27-35 address subject matter that was not described in the specification as enabled.

Applicants respond that enablement does not require that the precise language be repeated in the claims as used in the specification, although several claims have been amended to address Applicants' speculation that the Examiner is possibly concerned with a difference in terminology.

As explained at MPEP §2164.01, enablement is satisfied if the disclosure, when filed, contained sufficient information to enable one skilled in the art to make and use the claimed invention. Applicants respectfully submit that such enablement requirement is present, as explained below.

Relative to claim 27, Applicants have replaced "inserted into" with "set into", and believes this change addresses the Examiner's concern. Applicants further bring to the Examiner's attention the following descriptions:

Page 1, lines 13-14: "*Conventional marks and hypertext structures are static in both location and type, and comprise a tag which is hard-coded in the source file.*"

Page 1, lines 16-18: "*What is needed is a mechanism for "activemarks" which are dynamically set in a document using a programmable editor, and which provide the capability to dynamically link any pieces of text or locations in the edit view to any editor, or external, command.*"

Page 2, lines 1-3: "*The activemark structure utilizes a parsing mechanism which creates activemarks automatically as the user opens a document, and thereafter as the user enters information into the document, by incrementally parsing changes to the document.*"

It is noted that, however, this does not preclude hard-coding the activemarks as tags in a document being processed by an editor implementing our invention, as follows.

Page 3, lines 9-11: "*This editor may be, however, programmed to encode activemarks set up during the edit session as hard-coded tags in the document file.*"

Page 17, line 23-page 18, line 2: "*It will be understood that the live parser(s) and/or associated editor tools can generate any or all of these if so desired by the specific application, such as adding functionality-equivalent tags to the saved source document, or saving the activemarks in the file extended attributes or in a separate file for the next edit session.*"

Relative to claim 28, the Examiner alleges that the cited pages do not disclose a mark control module and the marks are solely as defined by a parsing editor.

Applicants submit that, in the present invention, the marks are indeed determined by the parser(s), but the actual code that maintains and handles the activemark data structures is a distinct control module, as follows.

Page 3, lines 17-24: "*In one aspect, the present invention provides a processing system for processing a document, the processing system comprises: (a) a programmable text processing module having means for loading the document and a parsing editor for initially parsing the document and thereafter incrementally parsing changes committed in the document; (b) a mark control module having means for setting a plurality of marks in the document, means for modifying the marks, and means for clearing the marks, and each of the marks comprising selected information with a command, the linking means being responsive to the operation of the parsing editor; ....*"

Page 3, line 25-page 4, line 2: "... *(d) an edit control module having means for controlling the text processing module, means for controlling the mark control module, and means for controlling the graphical user interface module.*"

Page 5, lines 5-7: "*Referring to Fig. 1, the edit system 10 comprises the following principal functional modules: an edit control module 11, a text processing module 12, an activemarks module 13, a graphical user interface (GUI) control module 14, and a commands/macro interface module 15.*"

Page 5, line 23 - Page 6, line 4: "*The activemarks module 13 is called to respond to the changes in the text, such as to adjust the activemarks to newly-changed text and to invoke exit points of activemarks affected by alterations to the document text. The activemarks module 13* *handles the processing of marks and activemarks including setting, clearing, changes to the marks resulting from modification to the text, setting and activation of activemark commands* and exit points. The functions of the activemarks module 13 are described in more detail below."

Page 8, lines 22-24: "*In response to selection of a particular key (or key combination), the activemarks module 13 in the edit system 10 will run the command(s) which are bound to the activemark(s) indicated.*"

The Examiner further says: "A "live parsing" is mentioned, however, a "parsing editor" is not mentioned.

In response, Applicants are uncertain whether the Examiner is merely making a grammatical quibble. Applicants submit that "parsing editor" should be understood, by one of skill in the art who is reading this application, to be an editor with associated/attached/running parsers.

Relative to claim 29, the Examiner states: "Applicant's cited pages mentions one or more "live parsers", however, does not mention "a plurality of parsing editors" providing a "unique functionality".

In response, Applicants submit that the Application makes clear that one or several live parser(s) and/or other editor external commands may be attached to run in the editor. Applicants additionally submit that it makes sense to those with ordinary skill in the art reading this application that any one of these provides its own unique functionality: why attach a few of the same kind?

Page 2, lines 15-17: "*According to the invention, the mark is set over, or linked to, a particular piece of text by a live parser and/or other external command(s) running off the editor, rather than being hard-coded in the text.*"

Page 7, lines 15-18: "*Instead of being "hard-coded" in the text of the document, the activemark according to the invention is set over a particular piece of text by a live parser and/or other external command(s) running in the edit system 10.*"

Page 7, line 24-page 8, line 6: "*One or more live parser(s) may be atached to the edit system 10 for a document. A parser first gets control after the file has been loaded into the edit system 10 (i.e. initial total parse), and thereafter it is called by the text processing module 12* *following changes made to the document (i.e., incremental parse). The live parser, which is typically an external command, parses the edit document. The live parser sets display attributes for the text elements in the edit view according to the syntax of the document (e.g. token highlightign), records classes in the edit system 10 with information regarding various constructs found in the document (e.g. C-language function bodies), and sets up activemarks accordingly."*

Page 17, lines 7-9: "*A feature of the activemark mechanism is that no change (such as tagging) is needed in the processed source file, nor its extended attributes, or in anothe file associated with it. All the functionality is handled by the live parser(s) manipulating the document via the activemarks mechanism.*"

The main example in the application refers to a C-language parser using activemarks. Another possible application example is mentioned:

Page 12, lines 1-4: "*As another example, an interactive demo of an application, or interactive navigation through a document, may be set through an appropriate source file in conjunction with a live parser and text formatter that interpret the text as menu items, table of contents, references to external commands, system utilities, etc.*"

Relative to claim 30, the Examiner alleges that the cited pages do not mention "a plurality of parsing editors" and "binding different actions to the same activemark set".

In response, Applicants submit that the Application makes clear the capability to bind different actions (commands) to the same activemark.

Page 1, lines 21-22: "*The activemark structure features the capability to bind commands to the mark.*"

Page 2, lines 17-19: "*Activemarks and their associated command(s) may also be set during the edit session by the user either manually, i.e., directly through the GUI (Graphical User Interface), or through action/tools provided by the GUI.*"

Page 8, lines 16-17: "*The user utilizes the QUERY MARK.COMMAND command to determine which command(s) are bound to an activemark.*"

Page 8, lines 22-24: "*In response to selection of a particular key (or key combination), the activemarks module 13 in the edit sysem 10 will run the command(s) which are bound to the* *activemark(s) indicated."*

Page 10, lines 14-19: "*The commands associated with the activemarks in the window 101 shown in Fig. 2 are set to call a function in the live parser which opens a source function-navigator window 103 as shown in Fig. 3 and runs the ocmmands associated with the particular activemark which has been activated. This feature serves to effectively chain activemarks in different windows (or levels of abstraction of the same source code) in the edit system 10.*"

Page 21, lines 11-13: "*When MARK RUN is invoked, the activemark command first modifies the mark's highlight for visual-feedback purposes, and then performs the rest of the operations associated with the mark.*"

Page 24, lines 9-11: "*The command SET MARKCOMMAND is provided to set the command binding of a mark in the document, turning the regular mark into an activemark, and the GET MARKCOMMAND is provided to get the command(s) bound to an activemark.*"

Page 24, lines 16-18: "*Once the command string is set for an activemark, the existing or pre-set command string may be queried, altered as needed, and set back for the activemark by another external editor utility or parser. In this manner, several commands may be chained off one activemark.*"

Relative to claims 31, 34, and 35, the Examiner states that the cited pages mention "using an HTML editor for HTML files" and "one or more live parsers" but fails to mention a "mark control module", a plurality of parsing editors and any other plurality of markup languages.

In response, Applicants point to the same comments above concerning different parsers and/or external commands and the mark control module.

Relative to claim 32, the Examiner states that the cited pages mention creating activemarks automatically when the user opens a document but fails to mention marks are defined dynamically by a parsing editor.

In response, Applicants submit that this feature is mentioned almost everywhere in the Application. As just two examples, the following two previously-recited passage are mentioned.

Page 2, lines 15-17: "*According to the invention, the mark is set over, or linked to, a particular piece of text by a live parser and/or other external command(s) running off the editor, rather than being hard-coded in the text.*"

Page 7, lines 15-18: "*Instead of being "hard-coded" in the,text of the document, the activemark according to the invention is set over a particular piece of text by a live parser and/or other external command(s) running in the edit system 10.*"

<u>Relative to claim 33</u>, as best understood, the Examiner raises a grammatical issue and have amended the claim language for clarity.

Accordingly, with the above clarification, Applicants respectfully request that the Examiner reconsider and withdraw the rejection for enablement.

## IV. THE PRIOR ART REJECTIONS

The Examiner alleges that the present invention defined by claims 1-16 and 21-35 is rendered obvious by IBM's "LPEX User's Guide and Reference", further in view of IBM Technical Disclosure Bulletin "Incrementally Imbedded Messages in an Edit View."

Applicants respectfully disagree.

It is first pointed out that development of the present invention did indeed, at least partially, rely upon IBM's LPEX tool as a convenient environment upon which to exemplarily implement the concepts of the invention. LPEX does <u>not</u>, however, <u>by itself</u> provide the capabilities described by the independent claims.

It is also noted that one of the co-inventors, Adrian Storisteanu, of the present invention is also a co-inventor of US Patent 6,185,591 (related to LPEX) and a co-author of one or more articles upon which the Examiner now considers as rendering obvious the present invention.

The specification, at lines 3-5 of page 2 and lines 1-3 of page 40, clearly indicates that the present invention can be implemented on the LPEX platform.

Relative to independent claims 1, 8, and 13, the Examiner states that the LPEX User's Guide teaches:

"*... a mark control module having means for setting a plurality of marks in the document, means for modifying said marks, and means for clearing said marks, and each of said marks comprising selected information in the document and said means for setting being responsive to the operation of said parsing editor without user intervention (LPEX on page 15 teaches setting* *marks on page 13 teaches the user can known how to highlight or mark a block of text and pages 17-18 teaches the parser uses colors to highlight (mark) different items in a programming language document, in other words, LPEX (parsing editor) has the capability of highlighting (marking) within a loaded document with the user intervention; wherein the user can modify the marks or unmarking the block of text);*"

In response, Applicants submit that LPEX teaches regular editor bookmarks. The distinction of the activemarks of the present invention over regular marks is clearly described in the description at lines 6-18 of page 1 of the present Application:

"*The concept of a mark is used in document processing systems, such as editors, to denote a label structure. In an editor, marks are typically used for setting bookmarks in the text....*

*The conventional concepts of "mark" and "hypertext" are restrictive in their definitions and, as a consequence, have limited functionality in the context of today's data processing systems. Conventional mark and hypertext structures are static in both location and type, and comprise a tag which is hard-coded in the source file. Beyond responding to the click of a mouse, conventional mark and hypertext structures offer no real interactivity.*

*What is needed is a mechanism for "activemarks" which are dynamically set in a document using a programmable editor, and which provide the capability to dynamically link any pieces of text or locations in the edit view to any editor, or external command.*"

Additionally, at lines 13-15 of page 8:

"*A user utilizes the SET MARK.COMMAND command to turn a conventional mark into an activemark, i.e., set the command binding of a mark. It will be appreciated that as such conventional marks in the edit system 10 are a subset of activemarks.*"

In the rejection currently of record, the Examiner points to pages 17-18 of the LPEX User's Guide, which describes that the parser uses colors to highlight text in the document. It appears that the Examiner is a bit confused by all the meanings of "mark" (e.g., as a verb/noun). In the present Application"marks/activemarks" are bookmarks. In contrast, "marking text" in LPEX means to select text (p. 13). "Setting a mark" in LPEX merely sets a conventional bookmark (p. 15). "Highlighting" by the parser is highlighting text in the document (pp. 17-18).

Finally, Applicants submit that marking languages, such as HTML, use special-meaning tags written into the document text itself, as described at the beginning of the specification in the above-recited section from page 1.

In the rejection, the Examiner continues:

*"... an edit control module having means for controlling said text processing module, means for controlling said mark control module, and means for controlling said graphical user interface module (LPEX on pages 17-18 teaches LPEX parser is an editor command that interactively works on the document)."*

In response, Applicants submit that LPEX does not teach or suggest any mark control module being controlled by the parser.

In the rejection, the Examiner continues:

*"... However, LPEX does not explicitly disclose "incrementally parsing changes committed in said document" and "linking said selected information with a command". IBM "Incrementally Imbedded Messages in a Edit View" on pages 1 and 4 teaches an incremental parser and pages 1 and 2 teaches a message is inserted into the edit view which refers to the text immediately above; the parser highlights the text in error and provides a message that describes the error; once the error is corrected, the parser re-parses and removes all messages, in other words, the messages can be commands or suggestions for correcting an error from the edit window. It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified IBM into LPEX to provide a way to incrementally parse a document and provide a message with each highlighted block of text, as taught by IBM, incorporated into the system of LPEX, in order to help users write programs in a Live Parsing Editing environment."*

In response, Applicants submit, first, that displaying embedded error messages in the document is not related in any manner whatsoever to the functionality, purpose, or scope of the activemarks of the present invention and certainly not related to linking marks with commands. As for the incremental parsing in LPEX, Applicants submit that this, again, is a basic editor functionality upon which the present invention has exemplarily been implemented.

Second, Applicants submit that the rejection currently of record fails to meet the burden of a *prima facie* rejection in that it merely alleges that the motivation to modify the primary reference is to obtain the benefit of having made the modification. Using this evaluation technique, <u>everything</u> would be obvious.

That is, the Examiner fails to point to any suggestion in the prior art references themselves to make a modification. Rather, the rejection merely alleges that a modification would provide a benefit, in contradiction of the clear guidelines in MPEP §2143.01: "*The mere fact that references <u>can</u> be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination.*" (Emphasis in MPEP itself)

Hence, turning to the clear language of the claims as defined by independent claim 1 (and substantially similarly by independent claims 8 and 13), there is no teaching or suggestion of "... *[a] processing system for processing a document, said processing system comprising:*

*a programmable text processing module having means for loading the document and a parsing editor for initially parsing the document and thereafter incrementally parsing changes committed in said document;*

*a <u>mark control module</u> having means for <u>setting</u> a plurality of marks in the document, means for <u>modifying</u> said marks, and means for <u>clearing</u> said marks, and each of said marks comprising selected information in the document and means for linking said selected information with a command, said linking means and said means for setting being responsive to the operation of said parsing editor without user intervention;*

*a graphical user interface module having means for displaying the document and means for controlling the display of the document; and*

*an <u>edit control module</u> having means for controlling said text processing module, means for <u>controlling said mark control module</u>, and means for controlling said graphical user interface module*" (emphasis Applicant's).

Relative to the rejection for claims 2, 14, and 15, Applicants again submit that displaying embedded error messages in the document is not related to the activemarks of the present invention. Moreover, Applicants submit that the Examiner's comment (e.g., "messages can be linked to each highlighted block of text") concerning the description on pages 1 and 3 of "Incrementally Imbedded Messages in a Edit View" misreads the reference. These messages are associated with the textual content of the document (they are generated by, e.g., spell-checkers, syntax checkers, compilers, etc.).

Relative to the rejection for claim 7, Applicants submit that LPEX does not teach associating marks with editor commands or macros. This is a significant novel feature of the present invention and non-trivial to implement. The fact that LPEX teaches invoking commands and macros is a basic feature (API) of the editor upon which the present invention is exemplarily implemented, as explained in the disclosure.

Relative to the rejection for claim 33, Applicants submit that what LPEX on page 13 teaches is simply text selection. The fact that it is called "block marking" in the LPEX product has nothing to with the marks/bookmarks/activemarks as used in the context of the present invention. It is also brought to the Examiner's attention that claim 33 recites "... other than static and hard coded ...."

Relative to the rejection for claim 35, the statement by the Examiner is not what is being claimed.

## V. FORMAL MATTERS AND CONCLUSION

The Examiner also objects to the drawings. Applicants will submit new formal drawings upon receiving indication of an allowance.

In view of the foregoing, Applicant submits that claims 1-16 and 21-38, all the claims presently pending in the application, are patentably distinct over the prior art of record and are in condition for allowance. The Examiner is respectfully requested to pass the above application to issue at the earliest possible time.

Should the Examiner find the application to be other than in condition for allowance, the Examiner is requested to contact the undersigned at the local telephone number listed below to discuss any other changes deemed necessary in a telephonic or personal interview.

The Commissioner is hereby authorized to charge any deficiency in fees or to credit any overpayment in fees to Assignee's Deposit Account No. 50-0510.

Respectfully Submitted,

Frederick Coop

Date: 11/12/04

Frederick E. Cooperrider
Reg. No. 36,769

**McGinn & Gibb, PLLC**
8321 Old Courthouse Rd. Suite 200
Vienna, VA 22182-3817
(703) 761-4100
**Customer No. 21254**

CERTIFICATION OF TRANSMISSION

I certify that I transmitted via facsimile to (703) 872-9306 this Amendment under 37 CFR §1.116 to Examiner Almari Romero Yuan on November 12, 2004.

Frederick Coop

Frederick E. Cooperrider
Reg. No. 36,769